# JASPERREPORTS - CROSSTABS

Crosstab (cross-tabulation) reports are reports containing tables that arrange data across rows and columns in a tabular form. Crosstab object is used for inserting a crosstab report within the main report. Crosstabs can be used with any level of data (nominal, ordinal, interval, or ratio), and usually display the summarized data, contained in report variables, in the form of a dynamic table. Variables are used to display aggregate data such as sums, counts, average values.

## Crosstab Properties

JRXML element **<crosstab>** is used to insert a crosstab into a report.

## Attribute

Following is a list of attribute of a **<crosstab>** element:

- **isRepeatColumnHeaders**: Indicates whether the column headers should be reprinted after a page break. The default value is *true*.
- **isRepeatRowHeaders**:Indicates whether the row headers should be reprinted after a crosstab column break. The default value is *true*.
- **columnBreakOffset**: When a column break occurs, indicates the amount of vertical space, measured in pixels, before the subsequent crosstab piece to be placed below the previous one on the same page. The default value is 10.
- **runDirection**: Indicates whether the crosstab data should be filled from left to right (LTR) or from right to left (RTL). The default value is LTR.
- **ignoreWidth**: Indicates whether the crosstab will stretch beyond the initial crosstab width limit and don't generate column breaks. Else it will stop rendering columns within the crosstab width limit and continue with the remaining columns only after all rows have started rendering. The default value is *false*.

## Sub Elements

A <crosstab> element has following sub elements :

- **<reportElement>**: This element defines the position, width, and height of the crosstab within its enclosing . Attributes for this element include all standard <reportElement> attributes.
- **<crosstabParameter>**: This element is used to access report variables and parameters from within the crosstab. Attributes for this element include:
    - *name*: This defines the parameter name.
    - *class*: This indicates the parameter class.
- **<parametersMapExpression>**: This element is used to pass a report variable or parameter containing an instance of *java.util.Map*, as a set of parameters for the crosstab. This element contains no attributes.
- **<crosstabDataset>**: This element defines the dataset to use to populate the crosstab (see next section for a detailed explanation). Attributes for this element include:

  - *isDataPreSorted*: This indicates whether the data in the dataset is pre-sorted. Default value is *false*.

- **<crosstabHeaderCell>**This element defines the content of the region found at the upper-left corner of the crosstab where column headers and row headers meet. The size of this cell is calculated automatically based on the defined row and column widths and heights.

- **<rowGroup>**: This element defines a group used to split the data into rows. Attributes for this element include:

  - *name*: This defines the name of the row group.
  - *width*: This defines the width of the row group.
  - *headerPosition*: This defines the position of the header contents (Top, Middle, Bottom, Stretch).
  - *totalPosition*: This defines the position of the entire column (Start, End, None).

  This element contains the following sub elements:

  - *<bucket>*
  - *<crosstabRowHeader>*
  - *<crosstabTotalRowHeader>*

- **<columnGroup>**: This element defines a group used to split the data into columns. Attributes for this element include:

  - *name*: This defines the column group name.
  - *height*: This defines the height of the column group header.
  - *headerPosition*: This defines the position of the header contents (*Right, Left, Center, Stretch*).
  - *totalPosition*: This defines the position of the entire column (*Start, End, None*).

  This element contains the following sub elements:

  - *<bucket>*
  - *<crosstabColumnHeader>*
  - *<crosstabTotalColumnHeader>*

- **<measure>**: This element defines the calculation to be performed across rows and columns. Attributes for this element include:

  - *name*: This defines the measure name.
  - *class*: This indicates the measure class.
  - *calculation*: This indicates the calculation to be performed between crosstab cell values. Its values could be any of these - *Nothing, Count, DistinctCount, Sum, Average, Lowest, Highest, StandardDeviation, Variance, First*. Default value is **Nothing**.

- **<crosstabCell>**: This element defines how data in non-header cells will be laid out. Attributes for this element include:

  - *columnTotalGroup*: This indicates the group to use to calculate the column total.

  - *height*: This defines the height of the cell.
  - *rowTotalGroup*: This indicates the group to use to calculate the row total.
  - *width*: This defines the width of the cell.

- **<whenNoDataCell>**: This element defines what to display on an empty crosstab cell. This element contains no attributes.

## Data Grouping in Crosstab

The crosstab calculation engine aggregates data by iterating through the associated dataset records. In order to aggregate data, one need to group them first. In a crosstab, rows and columns are based on specific group items, called **buckets**. A bucket definition should contain:

- *bucketExpression*: The expression to be evaluated in order to obtain data group items.
- *comparatorExpression*: Needed in the case the natural ordering of the values is not the best choice.
- *orderByExpression*: Indicates the value used to sort data.

Row and column groups (defined above) in a crosstab rely on **buckets**.

## Built-In Crosstab Total Variables

Below is a list of current value of measure and totals of different levels corresponding to the cell can be accessed through variables named according to the following scheme:

- The current value of a measure calculation is stored in a variable having the same name as the measure.
- *<Measure>_<Column Group>_ALL*: This yields the total for all the entries in the column group from the same row..
- *<Measure>_<Row Group>_ALL*: This yields the total for all the entries in the row group from the same column.
- *<Measure>_<Row Group>_<Column Group>_ALL*: This yields the combined total corresponding to all the entries in both row and column groups.

## Example

To demonstrate the crosstabs, let's write a new report template (jasper_report_template.jrxml). Here we will add the crosstab to summary section. Save it to the directory **C:\tools\jasperreports-5.0.1\test** directory. The contents of the file are as below:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE jasperReport PUBLIC "//JasperReports//DTD Report Design//EN"
"http://jasperreports.sourceforge.net/dtds/jasperreport.dtd">
<jasperReport xmlns="http://jasperreports.sourceforge.net/jasperreports"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://jasperreports.sourceforge.net/jasperreports
http://jasperreports.sourceforge.net/xsd/jasperreport.xsd"
name="jasper_report_template" language="groovy" pageWidth="595"
pageHeight="842" columnWidth="555" leftMargin="20" rightMargin="20"
topMargin="20" bottomMargin="20">
<parameter name="ReportTitle" />
<parameter name="Author" />
   <field name="name" />
   <field name="country" />
   <title>
       <band height="70">
```

```
          <line>
             <reportElement x="0" y="0" width="515"
             height="1"/>
          </line>
          <textField isBlankWhenNull="true" bookmarkLevel="1">
             <reportElement x="0" y="10" width="515"
             height="30"/>
             <textElement textAlignment="Center">
                <font size="22"/>
             </textElement>
             <textFieldExpression >
                <![CDATA[$P{ReportTitle}]]>
             </textFieldExpression>
             <anchorNameExpression>
                <![CDATA["Title"]]>
             </anchorNameExpression>
          </textField>
          <textField isBlankWhenNull="true">
             <reportElement  x="0" y="40" width="515" height="20"/>
             <textElement textAlignment="Center">
                  <font size="10"/>
             </textElement>
             <textFieldExpression >
                <![CDATA[$P{Author}]]>
             </textFieldExpression>
          </textField>
       </band>
    </title>
    <summary>
       <band height="60">
       <crosstab>
          <reportElement width="782" y="0" x="0" height="60"/>
           <rowGroup name="nameGroup" width="100">
          <bucket>
              <bucketExpression >
                 <![CDATA[$F{name}]]>
              </bucketExpression>
           </bucket>
           <crosstabRowHeader>
              <cellContents>
                 <box border="Thin" borderColor="black"/>
                 <textField>
                    <reportElement width="100" y="0" x="0" height="20"/>
                    <textElement textAlignment="Right"
                    verticalAlignment="Middle"/>
                    <textFieldExpression>
                       <![CDATA[$V{nameGroup}]]>
                    </textFieldExpression>
                 </textField>
              </cellContents>
              </crosstabRowHeader>

           </rowGroup>
           <columnGroup name="countryGroup" height="20">
             <bucket>
             <bucketExpression >
              $F{country}
              </bucketExpression>
              </bucket>
              <crosstabColumnHeader>
                <cellContents>
                 <box border="Thin" borderColor="black"/>
                 <textField isStretchWithOverflow="true">
                    <reportElement width="60" y="0" x="0" height="20"/>
                    <textElement verticalAlignment="Bottom"/>
                    <textFieldExpression>
                       <![CDATA[$V{countryGroup}]]>
                    </textFieldExpression>
                 </textField>
                </cellContents>
              </crosstabColumnHeader>

           </columnGroup>
           <measure name="tailNumCount"
           calculation="Count">
              <measureExpression>$F{country}</measureExpression>
           </measure>
           <crosstabCell height="20" width="60">
              <cellContents backcolor="#FFFFFF">
                 <box borderColor="black" border="Thin"/>
                 <textField>
                    <reportElement x="5" y="0" width="55" height="20"/>
                    <textElement textAlignment="Left"
                    verticalAlignment="Bottom"/>
                    <textFieldExpression
                       >
                       $V{tailNumCount}
                    </textFieldExpression>
                 </textField>
              </cellContents>
           </crosstabCell>
        </crosstab>
        </band>
 </summary>
 </jasperReport>
```

The details of the above file are as below:

- Crosstab is defined by the <crosstab> element.
- <rowGroup> element defines a group to split the data into rows. Here each row will display data for a different name.
- The <bucket> and <bucketExpression> elements define what report expression to use as a group delimiter for <rowGroup>. Here, we used the name field as a delimiter, in order to split the rows by name.
- The <crosstabRowHeader> element defines the expression to be used as a row header. It contains a single sub-element, namely <cellContents>, which acts like an inner band inside crosstab. Instead of defining variable name for the text field inside <crosstabRowHeader>, we have assigned the name to <rowGroup> (via its name attribute), hence it creates an implicit variable. The <crosstabRowHeader> element defines the contents of the header cell for the entire row. It takes a single <cellContents> element as its only sub-element.
- The <columnGroup> element as well as its sub-elements is analogous to the <rowGroup> element, except that it influences columns instead of rows.
- The <measure> element defines the calculation to be performed across rows and columns. The *calculation* attribute is set to *Count*.
- The <crosstabCell> element defines how data in non-header cells will be laid out. This element also contains a single <crosstabCell> element as its only sub-element.

The java codes for report filling remains unchanged. The contents of the file **C:\tools\jasperreports-5.0.1\test\src\com\tutorialspoint\JasperReportFill.java** are as below.

```
package com.tutorialspoint;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

import net.sf.jasperreports.engine.JRException;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.data.JRBeanCollectionDataSource;

public class JasperReportFill {
   @SuppressWarnings("unchecked")
   public static void main(String[] args) {
      String sourceFileName =
      "C://tools/jasperreports-5.0.1/test/jasper_report_template.jasper";

      DataBeanList DataBeanList = new DataBeanList();
      ArrayList<DataBean> dataList = DataBeanList.getDataBeanList();

      JRBeanCollectionDataSource beanColDataSource =
      new JRBeanCollectionDataSource(dataList);

      Map parameters = new HashMap();
      /**
       * Passing ReportTitle and Author as parameters
       */
      parameters.put("ReportTitle", "List of Contacts");
      parameters.put("Author", "Prepared By Manisha");

      try {
         JasperFillManager.fillReportToFile(
         sourceFileName, parameters, beanColDataSource);
      } catch (JRException e) {
         e.printStackTrace();
      }
   }
}
```

The contents of the POJO file **C:\tools\jasperreports-5.0.1\test\src\com\tutorialspoint\DataBean.java** are as below:

```
package com.tutorialspoint;

public class DataBean {
   private String name;
   private String country;

   public String getName() {
      return name;
   }

   public void setName(String name) {
      this.name = name;
   }

   public String getCountry() {
      return country;
   }

   public void setCountry(String country) {
      this.country = country;
   }
}
```

The contents of the file **C:\tools\jasperreports-5.0.1\test\src\com\tutorialspoint\DataBeanList.java** are as below:

```
package com.tutorialspoint;

import java.util.ArrayList;

public class DataBeanList {
   public ArrayList<DataBean> getDataBeanList() {
      ArrayList<DataBean> dataBeanList = new ArrayList<DataBean>();

      dataBeanList.add(produce("Manisha", "India"));
      dataBeanList.add(produce("Dennis Ritchie", "USA"));
      dataBeanList.add(produce("V.Anand", "India"));
      dataBeanList.add(produce("Shrinath", "California"));

      return dataBeanList;
   }

   /**
    * This method returns a DataBean object,
    * with name and country set in it.
    */
   private DataBean produce(String name, String country) {
      DataBean dataBean = new DataBean();
      dataBean.setName(name);
      dataBean.setCountry(country);
      return dataBean;
   }
}
```

## Report Generation

Next, let's compile and execute the above files using our regular ANT build process. The contents of the file build.xml (saved under directory C:\tools\jasperreports-5.0.1\test) are as below.

> *The import file - baseBuild.xml is picked from chapter [Environment Setup]() and should be placed in the same directory as the build.xml.*

```
<?xml version="1.0" encoding="UTF-8"?>
<project name="JasperReportTest" default="viewFillReport" basedir=".">
   <import file="baseBuild.xml" />
   <target name="viewFillReport"
      depends="compile,compilereportdesing,run"
      description="Launches the report viewer to preview
      the report stored in the .JRprint file.">
      <java classname="net.sf.jasperreports.view.JasperViewer"
      fork="true">
         <arg value="-F${file.name}.JRprint" />
         <classpath ref />
      </java>
   </target>
   <target name="compilereportdesing"
      description="Compiles the JXML file and
      produces the .jasper file.">
      <taskdef name="jrc"
      classname="net.sf.jasperreports.ant.JRAntCompileTask">
         <classpath ref />
      </taskdef>
      <jrc destdir=".">
         <src>
         <fileset dir=".">
            <include name="*.jrxml" />
         </fileset>
         </src>
         <classpath ref />
      </jrc>
   </target>
</project>
```

Next, let's open command line window and go to the directory where build.xml is placed. Finally execute the command **ant -Dmain-class=com.tutorialspoint.JasperReportFill** (viewFullReport is the default target) as follows:

```
C:\tools\jasperreports-5.0.1\test>ant -Dmain-class=com.tutorialspoint.JasperReportFill
Buildfile: C:\tools\jasperreports-5.0.1\test\build.xml

clean-sample:
   [delete] Deleting directory C:\tools\jasperreports-5.0.1\test\classes
   [delete] Deleting: C:\tools\jasperreports-5.0.1\test\jasper_report_template.jasper

compile:
    [mkdir] Created dir: C:\tools\jasperreports-5.0.1\test\classes
    [javac] C:\tools\jasperreports-5.0.1\test\baseBuild.xml:28:
    warning: 'includeantruntime' was not set, defaulting to
    [javac] Compiling 3 source files to C:\tools\jasperreports-5.0.1\test\classes

compilereportdesing:
      [jrc] Compiling 1 report design files.
      [jrc] log4j:WARN No appenders could be found for logger
      (net.sf.jasperreports.engine.xml.JRXmlDigesterFactory).
      [jrc] log4j:WARN Please initialize the log4j system properly.
      [jrc] log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
      [jrc] File : C:\tools\jasperreports-5.0.1\test\jasper_report_template.jrxml ... OK.

run:
     [echo] Runnin class : com.tutorialspoint.JasperReportFill
     [java] log4j:WARN No appenders could be found for logger
     (net.sf.jasperreports.extensions.ExtensionsEnvironment).
     [java] log4j:WARN Please initialize the log4j system properly.

viewFillReport:
     [java] log4j:WARN No appenders could be found for logger (
     net.sf.jasperreports.extensions.ExtensionsEnvironment).
     [java] log4j:WARN Please initialize the log4j system properly.

BUILD SUCCESSFUL
Total time: 20 minutes 53 seconds
```

As a result of above compilation, a JasperViewer window opens up as in the screen below:

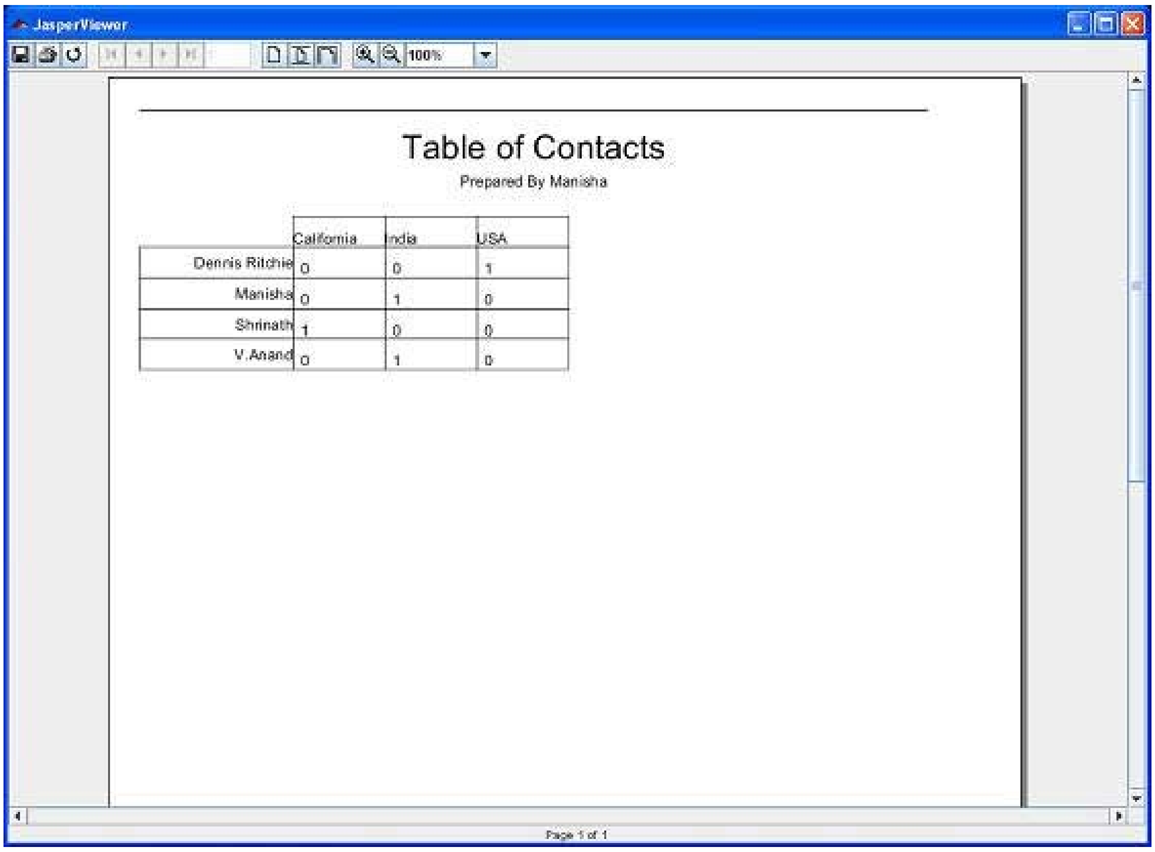

Table of Contents

Prepared By Manisha

| | California | India | USA |
|---|---|---|---|
| Dennis Ritchie | 0 | 0 | 1 |
| Manisha | 0 | 1 | 0 |
| Shrinath | 1 | 0 | 0 |
| V.Anand | 0 | 1 | 0 |

Here we see that the each country and name are tabulated.